09/803,168

S141	57	lock$3 and S138	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/03/18 10:31
S142	10	thread$1 with (creat$3 or generat$4) with object$1 same (indicat$4 or flag$3 or label$2) same own$7	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/03/18 11:36
S143	0	("6772153").URPN.	USPAT	OR	OFF	2005/03/18 10:47
S144	10	("4104718" | "5907675" | "5918229" | "5931919" | "5996032" | "6023700" | "6314563" | "6389452" | "6415334" | "6473820").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/18 10:47
S145	10	("4104718" | "5907675" | "5918229" | "5931919" | "5996032" | "6023700" | "6314563" | "6389452" | "6415334" | "6473820").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/18 10:47
S146	10	thread$1 with (creat$3 or generat$4) with object$1 same (indicat$4 or flag$3 or label$2 or bit$1) same own$7	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/03/18 11:36
S147	134	thread$1 with (creat$3 or generat$4) with object$1 same (indicat$4 or flag$3 or label$2)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/03/18 11:36
S148	142	thread$1 with (creat$3 or generat$4) with object$1 same (indicat$4 or flag$3 or label$2 or bit$1)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/03/18 13:12
S149	15	thread$1 with (creat$3 or generat$4) with object$1 same (indicat$4 or flag$3 or label$2 or bit$1) same lock$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/03/18 13:22
S150	9	(thread$1 with (creat$3 or generat$4) with object$1 same (indicat$4 or flag$3 or label$2 or bit$1) same own$3) and lock$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/03/18 13:23
S151	9	(thread$1 with (creat$3 or generat$4) with object$1 same (indicat$4 or flag$3 or label$2 or bit$1) same own$6) and lock$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/03/18 13:23
S152	1	"6883026".pn.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/28 17:33

S153	10	(KAWACHIYA near KIYOKUNI).in.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/28 17:35
S154	653	(flag or bit or indicat$4) same (own$3 or local$4 or refer$4) same object$1 same thread	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/28 17:38
S155	4742	718/1-107.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/28 17:38
S156	99	S154 and S155	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/28 17:38
S157	87	S156 and @ay <="2001"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/28 17:38
S158	1	"5323519"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 11:27
S159	536	707/206.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 14:02
S160	653	(flag or bit or indicat$4) same (own$3 or local$4 or refer$4) same object$1 same thread	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 14:06
S161	4742	718/1-107.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 14:03
S162	25324	"711"/$.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 14:04
S163	171	(S159 or S161 or S162) and S160	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 14:04
S164	7	(S159 or S162) and S161 and S160	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 14:04

S165	102	((flag or bit or indicat$4) same (own$3 or local$4 or refer$4) same object$1 same thread) and (garbage near collector$1)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 14:07
S166	40	((flag or bit or indicat$4) same (own$3 or local$4 or refer$4) near5 thread same object$1) and (garbage near collector$1)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 15:37
S167	1	"6912553".pn.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 15:46
S168	1	"6487714".pn.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 16:01
S169	0	"09356532".pn.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 16:01
S170	0	("not" near2 lock) with (flag or bit)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:28
S171	0	("not" near2 lock$3) same (flag or bit)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:28
S172	589	set near5 (flag or bit or indicator) with thread	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:29
S173	3488	set near5 (flag or bit or indicator) with (thread or lock$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:29
S174	2077	set near2 (flag or bit or indicator) with (thread or lock$1)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:30
S175	0	"not" near2 set near2 (flag or bit or indicator) with (thread or lock$1)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:31
S176	0	"not" near2 set near2 (flag or bit or indicator) near (thread or lock$1)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:31

S17 7	1172	set near2 (flag or bit or indicator) near2 (thread or lock$1)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:32
S17 8	2983	garbage near collection	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:32
S17 9	4086	garbage near collect$4	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:32
S18 0	48	S177 and S179	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:44
S18 1	60	flag with lock$3 with thread$1	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/06/29 16:52
S18 2	38	(den$4 or reject$3) near2 lock$3 with (flag or bit or indicator)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 17:02
S18 3	165	(reject$3 or den$4) with access$3 same thread$1	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 17:03
S18 4	183481	set$4 near5 (flag or bit or indicator)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 17:04
S18 5	44	S183 and S184	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/06/29 17:04

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: [✓ The ACM Digital Library] The Guide

thread local object flag bit indicator

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **thread** **local** **object** **flag** **bit** **indicator**

Found **38,567** of **157,956**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Implementation of Argus

B. Liskov, D. Curtis, P. Johnson, R. Scheifer

November 1987 **ACM SIGOPS Operating Systems Review , Proceedings of the eleventh ACM Symposium on Operating systems principles**, Volume 21 Issue 5

Full text available: pdf(1.34 MB) Additional Information: full citation, abstract, references, citings, index terms

Argus is a programming language and system developed to support the construction and execution of distributed programs. This paper describes the implementation of Argus, with particular emphasis on the way we implement atomic actions, because this is where Argus differs most from other implemented systems. The paper also discusses the performance of Argus. The cost of actions is quite reasonable, indicating that action systems like Argus are practical.

2 Making operating systems more robust: RacerX: effective, static detection of race conditions and deadlocks

Dawson Engler, Ken Ashcraft

October 2003 **Proceedings of the nineteenth ACM symposium on Operating systems principles**

Full text available: pdf(310.63 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes RacerX, a static tool that uses flow-sensitive, interprocedural analysis to detect both race conditions and deadlocks. It is explicitly designed to find errors in large, complex multithreaded systems. It aggressively infers checking information such as which locks protect which operations, which code contexts are multithreaded, and which shared accesses are dangerous. It tracks a set of code features which it uses to sort errors both from most to least severe. It uses novel ...

Keywords: deadlock detection, program checking, race detection

3 Special issue on prototypes of deductive database systems: The aditi deductive database system

Jayen Vaghani, Kotagiri Ramamohanarao, David B. Kemp, Zoltan Somogyi, Peter J. Stuckey, Tim S. Leask, James Harland

April 1994 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 3 Issue 2

Full text available: pdf(2.67 MB) Additional Information: full citation, abstract, references, citings

Deductive databases generalize relational databases by providing support for recursive views and non-atomic data. Aditi is a deductive system based on the client-server model; it is inherently multi-user and capable of exploiting parallelism on shared-memory multiprocessors. The back-end uses relational technology for efficiency in the management of disk-based data and uses optimization algorithms especially developed for the bottom-up evaluation of logical queries involving recursion. The front ...

Keywords: implementation, logic, multi-user, parallelism, relational database

4 A regular architecture for operating system
Vadim G. Antonov
July 1990 **ACM SIGOPS Operating Systems Review**, Volume 24 Issue 3
Full text available: pdf(1.07 MB) Additional Information: full citation, abstract, index terms

This paper describes an object-oriented operating system with regular architecture and minimal functionally complete set of primitives. It also considers the possibility of implementation of efficient system with this architecture on von Neumann machines.

5 Implementing an on-the-fly garbage collector for Java
Tamar Domani, Elliot K. Kolodner, Ethan Lewis, Eliot E. Salant, Katherine Barabash, Itai Lahan, Yossi Levanoni, Erez Petrank, Igor Yanorer
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management**, Volume 36 Issue 1
Full text available: pdf(1.33 MB) Additional Information: full citation, abstract, citings, index terms

Java uses garbage collection (GC) for the automatic reclamation of computer memory no longer required by a running application. GC implementations for Java Virtual Machines (JVM) are typically designed for single processor machines, and do not necessarily perform well for a server program with many threads running on a multiprocessor. We designed and implemented an on-the-fly GC, based on the algorithm of Doligez, Leroy and Gonthier [13, 12] (DLG), for Java in this environment. An *on-the-f* ...

***Keywords**: Java, concurrent garbage collection, garbage collection, memory management, on-the-fly garbage collection, programming languages*

6 First-class user-level threads
Brian D. Marsh, Michael L. Scott, Thomas J. LeBlanc, Evangelos P. Markatos
September 1991 **ACM SIGOPS Operating Systems Review , Proceedings of the thirteenth ACM symposium on Operating systems principles**, Volume 25 Issue 5
Full text available: pdf(1.53 MB) Additional Information: full citation, abstract, references, citings, index terms

It is often desirable, for reasons of clarity, portability, and efficiency, to write parallel programs in which the number of processes is independent of the number of available processors. Several modern operating systems support more than one process in an address space, but the overhead of creating and synchronizing kernel processes can be high. Many runtime environments implement lightweight processes (threads) in user space, but this approach usually results in second-class status for threa ...

7 Balancing performance and flexibility with hardware support for network architectures
Ilija Hadžić, Jonathan M. Smith
November 2003 **ACM Transactions on Computer Systems (TOCS)**, Volume 21 Issue 4

Full text available: pdf(719.03 KB) Additional Information: full citation, abstract, references, index terms

The goals of performance and flexibility are often at odds in the design of network systems. The tension is common enough to justify an architectural solution, rather than a set of context-specific solutions. The Programmable Protocol Processing Pipeline (P4) design uses programmable hardware to selectively accelerate protocol processing functions. A set of field-programmable gate arrays (FPGAs) and an associated library of network processing modules implemented in hardware are augmented with so ...

Keywords: FPGA, P4, computer networking, flexibility, hardware, performance, programmable logic devices, programmable networks, protocol processing

8 Automatic reconfiguration in Autonet
Thomas L. Rodeheffer, Michael D. Schroeder
September 1991 **ACM SIGOPS Operating Systems Review , Proceedings of the thirteenth ACM symposium on Operating systems principles**, Volume 25 Issue 5

Full text available: pdf(1.90 MB) Additional Information: full citation, abstract, references, citings, index terms

Autonet is a switch-based local area network using 100 Mbit/s full-duplex point-to-point links. Crossbar switches are interconnected to other switches and to host controllers in an arbitrary pattern. Switch hardware uses the destination address in each packet to determine the proper outgoing link for the next step in the path from source to destination. Autonet automatically recalculates these forwarding paths in response to failures and additions of network components. This automatic reconfigur ...

9 Special issue: Game-playing programs: theory and practice
M. A. Bramer
April 1972 **ACM SIGART Bulletin**, Issue 80

Full text available: pdf(9.23 MB) Additional Information: full citation, abstract

This collection of articles has been brought together to provide SIGART members with an overview of Artificial Intelligence approaches to constructing game-playing programs. Papers on both theory and practice are included.

10 Industrial sessions: big data: The SDSS skyserver: public access to the sloan digital sky server data
Alexander S. Szalay, Jim Gray, Ani R. Thakar, Peter Z. Kunszt, Tanu Malik, Jordan Raddick, Christopher Stoughton, Jan vandenBerg
June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data**

Full text available: pdf(1.48 MB) Additional Information: full citation, abstract, references, citings, index terms

The SkyServer provides Internet access to the public Sloan Digital Sky Survey (SDSS) data for both astronomers and for science education. This paper describes the SkyServer goals and architecture. It also describes our experience operating the SkyServer on the Internet. The SDSS data is public and well-documented so it makes a good test platform for research on database algorithms and performance.

11 High-performance Java codes for computational fluid dynamics
Christopher Riley, Siddhartha Chatterjee, Rupak Biswas
June 2001 **Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande**

Full text available: pdf(831.27 KB) Additional Information: full citation, abstract, references, index terms

The computational science community is reluctant to write large-scale computationally-intensive applications in Java due to concerns over Java's poor performance, despite the claimed software engineering advantages of its object-oriented features. Naive Java implementations of numerical algorithms can perform poorly compared to corresponding Fortran or C implementations. To achieve high performance, Java applications must be designed with good performance as a primary goal. This paper present ...

12 Bifurcated routing in computer networks
Wai Sum Lai
July 1985 **ACM SIGCOMM Computer Communication Review**, Volume 15 Issue 3
Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references

This paper presents a characterization and a survey of multiple path routing in computer networks. It also develops a routing protocol that achieves load sharing and combines the strengths of both virtual circuit and datagram networks.

13 Speculative synchronization: applying thread-level speculation to explicitly parallel applications
José F. Martínez, Josep Torrellas
October 2002 **Proceedings of the 10th international conference on Architectural support for programming languages and operating systems**, Volume 36 , 30 , 37 Issue 5 , 5 , 10
Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, citings

Barriers, locks, and flags are synchronizing operations widely used programmers and parallelizing compilers to produce race-free parallel programs. Often times, these operations are placed suboptimally, either because of conservative assumptions about the program, or merely for code simplicity.We propose *Speculative Synchronization,* which applies the philosophy behind Thread-Level Speculation (TLS) to explicitly parallel applications. Speculative threads execute past active barriers, busy ...

14 Scheduler-conscious synchronization
Leonidas I. Kontothanassis, Robert W. Wisniewski, Michael L. Scott
February 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 1
Full text available: pdf(682.20 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Efficient synchronization is important for achieving good performance in parallel programs, especially on large-scale multiprocessors. Most synchronization algorithms have been designed to run on a dedicated machine, with one application process per processor, and can suffer serious performance degradation in the presence of multiprogramming. Problems arise when running processes block or, worse, busy-wait for action on the part of a process that the scheduler has chosen not to run. We show ...

Keywords: barriers, busy-waiting, kernel-user interaction, locks, mutual exclusion, preemption, scalability, scheduling, synchronization

15 Curriculum '78: recommendations for the undergraduate program in computer science— a report of the ACM curriculum committee on computer science
Richard H. Austing, Bruce H. Barnes, Della T. Bonnette, Gerald L. Engel, Gordon Stokes
March 1979 **Communications of the ACM**, Volume 22 Issue 3
Full text available: pdf(2.20 MB) Additional Information: full citation, abstract, references, citings

Contained in this report are the recommendations for the undergraduate degree program in Computer Science of the Curriculum Committee on Computer Science (C3S) of the

Association for Computing Machinery (ACM). The core curriculum common to all computer science undergraduate programs is presented in terms of elementary level topics and courses, and intermediate level courses. Elective courses, used to round out an undergraduate program, are then discussed, and ...

Keywords: computer science curriculum, computer science education, computer science undergraduate degree programs, computer sciences courses, continuing education, service courses

16 Principles of database buffer management
Wolfgang Effelsberg, Theo Haerder
December 1984 **ACM Transactions on Database Systems (TODS)**, Volume 9 Issue 4

Full text available: pdf(2.39 MB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper discusses the implementation of a database buffer manager as a component of a DBMS. The interface between calling components of higher system layers and the buffer manager is described; the principal differences between virtual memory paging and database buffer management are outlined; the notion of referencing versus addressing of database pages is introduced; and the concept of fixing pages in the buffer to prevent uncontrolled replacement is explained. Three basic t ...

17 An on-the-fly mark and sweep garbage collector based on sliding views
Hezi Azatchi, Yossi Levanoni, Harel Paz, Erez Petrank
October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications**, Volume 38 Issue 11

Full text available: pdf(244.12 KB) Additional Information: full citation, abstract, references, citings, index terms

With concurrent and garbage collected languages like Java and C# becoming popular, the need for a suitable non-intrusive, efficient, and concurrent multiprocessor garbage collector has become acute. We propose a novel mark and sweep on-the-fly algorithm based on the sliding views mechanism of Levanoni and Petrank. We have implemented our collector on the Jikes Java Virtual Machine running on a Netfinity multiprocessor and compared it to the concurrent algorithm and to the stop-the-world collecto ...

Keywords: concurrent garbage collection, garbage collection, memory management, on-the-fly garbage collection, runtime systems

18 gIBIS: a hypertext tool for exploratory policy discussion
Jeff Conklin, Michael L. Begeman
October 1988 **ACM Transactions on Information Systems (TOIS)**, Volume 6 Issue 4

Full text available: pdf(2.23 MB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper describes an application-specific hypertext system designed to facilitate the capture of early design deliberations. It implements a specific method, called Issue Based Information Systems (IBIS), which has been developed for use on large, complex design problems. The hypertext system described here, gIBIS (for graphical IBIS), makes use of color and a high-speed relational database server to facilitate building and browsing typed IBIS networks. Further, gIBIS is designed to supp ...

19
Resource partitioning in general purpose operating systems: experimental results in

Windows NT
D. G. Waddington, D. Hutchison
October 1999 **ACM SIGOPS Operating Systems Review**, Volume 33 Issue 4
Full text available: pdf(1.56 MB) Additional Information: full citation, abstract, index terms

The principal role of the operating system is that of resource management. Its task is to present a set of appropriate services to the applications and users it supports. Traditionally, general-purpose operating systems, including Windows NT, federate resource sharing in a fair manner, with the predominant goal of efficient resource utilisation. As a result the chosen scheduling algorithms are not suited to applications that have stringent Quality-of-Service (QoS) and resource management require ...

20 Cheap recovery: a key to self-managing state
Andrew C. Huang, Armando Fox
February 2005 **ACM Transactions on Storage (TOS)**, Volume 1 Issue 1
Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, references, index terms

Cluster hash tables (CHTs) are key components of many large-scale Internet services due to their highly-scalable performance and the prevalence of the type of data they store. Another advantage of CHTs is that they can be designed to be as self-managing as a cluster of stateless servers. One key to achieving this extreme manageability is reboot-based recovery that is predictably fast and has modest impact on system performance and availability. This "cheap" recovery mechanism simplifies manageme ...

Keywords: Cluster hash table, manageability, quourum replication, storage systems design

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player